Comparative Study of Methods for Parsing HTTPS Access Data in Localized Electronic Resource Usage Statistics Systems: Postprint

Authors: Chen Guang

Date: 2023-07-26T00:00:00+00:00

Abstract

[Purpose/Significance]To address the emerging challenges faced by localized electronic resource usage statistics systems, this study proposes methods for parsing HTTPS access data and conducts analysis and evaluation, providing reference for libraries to address the data collection challenges of HTTPS-based electronic resource access in localized usage statistics systems.[Method/Process]From four aspects including hardware/software requirements, network conditions, system functionality, and user cooperation requirements, three methods are compared and evaluated: browser extensions, proxy programs supporting man-in-the-middle technology, and gateway devices supporting SSL proxy.[Results/Conclusion]Research shows that proxy programs supporting man-in-the-middle technology have moderate costs and the strongest system functionality, making them most suitable for application in localized electronic resource usage statistics systems. Beyond solving the HTTPS data collection issue, ensuring user privacy and data security while obtaining user cooperation represents the greatest challenge in implementing localized electronic resource usage statistics systems.

Full Text

Preamble

Vol. 63 No. 14 July 2019 ChinaXiv Cooperative Journal

A Comparative Study of Methods for Parsing HTTPS Access Data in Localized Electronic Resource Usage Statistics Systems

Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, Fuzhou 350002

Abstract:

[**Purpose/Significance**] Aiming to address new challenges facing localized electronic resource usage statistics systems, this paper proposes methods for parsing HTTPS access data, analyzes and evaluates these methods, and provides a reference for libraries to solve the collection problem of electronic resource access data based on the HTTPS protocol in localized usage statistics systems. [**Method/Process**] From four aspects—hardware and software requirements, network conditions, system functionality, and user cooperation needs—this paper compares and evaluates three methods: browser extensions, proxy programs supporting man-in-the-middle technology, and gateway devices supporting SSL proxy. [**Result/Conclusion**] The study shows that proxy programs supporting man-in-the-middle technology offer moderate cost and the strongest system functionality, making them most suitable for localized electronic resource usage statistics systems. On the basis of solving HTTPS access data collection, how to ensure user privacy and data security while obtaining user cooperation and coordination represents the greatest difficulty in applying localized electronic resource usage statistics systems.

Classification Number: G250.7
Keywords: electronic resources; usage statistics; browser extension; proxy program; SSL proxy gateway
DOI: 10.13266/j.issn.0252-3116.2019.14.005

Electronic resources constitute the core of current library collections, with their content continuously expanding and subscription fees rising annually. Under increasing budget pressure, libraries must effectively evaluate electronic resources to inform procurement decisions and maximize the rational use of limited funds. In the evaluation system for electronic resources, usage statistics serve as a crucial metric. These statistics accurately reflect the utilization of electronic resources, providing important references for libraries to restructure web portals, develop user training programs, identify key electronic resource products, and assist librarians in making collection development decisions regarding electronic resource acquisition and management.

Currently, libraries primarily obtain electronic resource usage statistics through Counter-compliant reports provided by database vendors, using independent statistical software such as ScholarlyStats and ExLibris UStat or electronic resource management systems to automate data collection and statistical work. Although vendor-provided Counter reports are well-developed and convenient,

they have certain limitations: (1) Vendor statistics sometimes fail to reflect actual user behavior, as user errors, repeated refreshes, and other actions can cause statistical usage to differ from actual usage. When usage rates are low, database vendors may not provide accurate figures. (2) Vendor reports only provide statistical numbers, which cannot meet libraries' needs for timely, in-depth analysis and data mining at the content or user level. (3) Vendor usage reports cannot respond to libraries' self-built systems such as collection management systems, institutional repositories, and scientific databases. (4) Counter reports are not generated in real-time, failing to meet libraries' needs for real-time usage statistics.

To address these limitations and satisfy libraries' needs for in-depth analysis and data mining of electronic resources, domestic libraries have successively conducted research on localized electronic resource usage statistics systems and achieved fruitful results. However, in recent years, localized systems face a new situation: an increasing number of databases have shifted from HTTP-based access to HTTPS-based access to ensure data security. For example, the OSA database deployed HTTPS-based access in 2017, while ACS and Wiley did so in 2018, discontinuing HTTP-based access. ScienceDirect, Nature, and Springer had already adopted HTTPS-based access before 2017. Since HTTP transmits data in plaintext while HTTPS uses encrypted transmission, localized electronic resource usage statistics systems originally adapted for HTTP can no longer collect access data after the protocol change. How to collect access data for electronic resources based on the HTTPS protocol has become an urgent problem for localized statistics systems.

This paper proposes three methods for parsing HTTPS access data, comprehensively compares their advantages and disadvantages across four dimensions—hardware/software requirements, network conditions, system capabilities, and user cooperation needs—and provides references for libraries to solve the collection problem of HTTPS-based electronic resource access data.

2 Background Analysis

2.1 Research Status of Localized Electronic Resource Usage Statistics Systems

Domestic libraries began researching localized electronic resource usage statistics systems early on. Technically, these systems can be divided into two main types: gateway log-based collection and analysis mode, and bypass monitoring-based collection and analysis mode.

Gateway log-based research includes: building electronic resource usage statistics systems using gateway logs; constructing electronic journal database statistical analysis systems by mining firewall logs; building electronic resource log statistics systems through Web logs from proxy servers; and log processing and data mining for library digital resource access systems. Bypass monitoring-based research includes: studying library user information behavior data col-

lection methods based on the ERU system; designing and applying electronic resource access management and control systems using bypass monitoring; designing and applying university electronic resource access management and control systems using bypass monitoring; and designing and implementing digital resource evaluation systems based on bypass monitoring.

In the gateway log-based collection and analysis mode, gateway devices such as core switches, firewalls, and proxy servers record Internet access data and generate log files containing electronic resource access data. In this case, libraries can implement certain log harvesting strategies, write log harvesting programs, filter and analyze log information, and generate usage statistics reports. The advantage of this mode is that it requires no changes to existing network topology or additional hardware devices, as it directly uses the log functions built into gateway devices. The limitation is that gateway logs may not contain complete information, failing to meet libraries' needs for in-depth data analysis and mining. The generation frequency of usage reports depends on log file generation frequency and log harvesting program performance, making real-time report generation difficult and preventing real-time monitoring of electronic resource usage. Since this method only collects log information, it cannot promptly terminate violations when users engage in unauthorized electronic resource usage.

The bypass monitoring-based collection and analysis mode adds a dedicated data analysis server to the existing network topology, connecting it to the gateway device at the network exit. Through port mirroring on the gateway device, data packets are copied to the data analysis server, which captures and parses them, filters and analyzes the content, and generates electronic resource usage statistics reports. The advantage of this mode is that using port mirroring to copy data packets requires no changes to the original network topology and does not affect user access behavior. It enables real-time monitoring of electronic resource usage for violation warnings and provides complete and accurate electronic resource usage data that meets libraries' deep mining needs. The limitation is the need for a dedicated data analysis server for monitoring, collection, and analysis, resulting in higher costs. Although it can monitor violations in real-time, the bypass monitoring approach does not participate in user access behavior, so violations cannot be immediately terminated when they occur.

2.2 HTTP and HTTPS

The Hypertext Transfer Protocol (HTTP) is the communication protocol used on the Internet. HTTP is a stateless, simple, fast, and reliable TCP-based transmission protocol, primarily applied for duplex communication between Web browsers and Web servers. Currently, most Web servers on the Internet use HTTP to transmit data, and most electronic resource access is HTTP-based. Although HTTP is convenient and fast, it has data security issues. HTTP transmits data in plaintext, making transmitted data transparent to all network devices along the transmission path, which allows third parties to eavesdrop on or tamper with data or even impersonate Web servers to communicate with users.

To address HTTP's data security issues, Netscape designed the SSL (Secure Sockets Layer) protocol to encrypt HTTP-transmitted data and applied SSL to its browser, thus creating HTTPS. The SSL protocol has three versions, with SSL 3.0 being the latest. In 1999, the Internet standardization organization ISO/IEC took over from Netscape and released TLS 1.0, an upgraded version of SSL. TLS has undergone two upgrades, with the latest being the TLS 1.2 revision released in 2011. SSL and its successor TLS are security protocols that provide security and data integrity for network communication, with main functions including: authenticating users and servers to ensure data is sent to correct clients and servers; encrypting data to prevent theft during transmission; and maintaining data integrity to ensure data is not altered during transmission.

As shown in Figure 1 [Figure 1: see original paper], the biggest difference between HTTP and HTTPS is that HTTPS introduces a security layer in HTTP to encrypt data, using either SSL or TLS protocols. Data is encrypted before reaching the transport layer, and all data transmitted across the network is encrypted, preventing data modification and tampering. HTTPS's added security layer makes it more costly to deploy than HTTP, requiring more server resources and longer access times. Despite higher deployment requirements, more electronic resource providers are gradually adopting HTTPS-based access to replace HTTP-based access for their data security.

3 Problems and Solutions for Localized Electronic Resource Usage Statistics Systems

For electronic resources accessed via HTTPS, whether using gateway log-based or bypass monitoring-based modes, access data reaching the gateway or data analysis server is already encrypted ciphertext. This prevents gateways or data analysis servers from obtaining detailed content, allowing only three pieces of information: user IP address, server IP address, and domain name—insufficient for generating electronic resource usage statistics reports. Existing localized electronic resource usage statistics systems are no longer applicable to the new electronic resource access method and urgently need to solve the collection problem for electronic resource usage data based on HTTPS protocol.

From the HTTPS working principle perspective, decrypting HTTPS can be accomplished in two ways: (1) HTTPS encryption and decryption occur at the security layer; for the application layer, data is either not yet encrypted or already decrypted. Placing monitoring programs at the application layer can obtain unencrypted data. (2) Using man-in-the-middle (MITM) technology to control data communication between client and server. Adding a third party between client and server, this third party establishes connections with both the real server (acting as client) and real client (acting as server), making both communication endpoints believe they are talking directly to each other while the entire session is actually controlled by the third party.

3.1 Browser Extension Programs

For the first method, browsers are the primary tools for accessing Web resources and operate at the application layer. All network access data is transparent to browsers and can be viewed in plaintext. By monitoring browser communication data, HTTPS-based access data can be obtained. This method essentially bypasses the security layer to monitor data at the application layer.

Mainstream browsers support extension programs, which are codes that modify Web browser functions using standard Web technologies (JavaScript, HTML, and CSS) and specialized JavaScript APIs to implement network request control and various event monitoring functions. Table 1 shows the names, kernel information, and supported extension interfaces of mainstream browsers.

Browsers in Table 1 use four kernels: Trident, WebKit, Blink, and Quantum. Extensions for browsers with the same kernel are mutually compatible. Blink is an upgraded version of WebKit, and extensions for these two kernels are also compatible. Firefox Quantum's WebExtensions API is compatible with Chrome Extension, so when developing browser extensions, only BHO and Chrome Extension need to be written to be compatible with mainstream browsers. Taking WebExtensions API as an example, Firefox Quantum's official documentation shows that the WebRequest module provides onBeforeRequest (triggered when browser sends request) and onCompleted (triggered when browser request completes) methods. By adding monitoring code to these two methods, creating corresponding extensions, and installing them in users' browsers, when users access electronic resources, request information and server response content can be simultaneously sent to the library's server, achieving collection of HTTPS-based electronic resource access data.

3.2 Man-in-the-Middle Technology

There are two methods to decrypt HTTPS using MITM technology. The first uses SSL proxy-supported gateway devices to replace original gateways or add them to the existing network topology. SSL proxy-supported gateway devices use SSL proxy certificates to replace encrypted Web site certificates, sending SSL proxy certificates to client Web browsers. During this process, the device acts as both SSL client and SSL server to establish SSL connections with Web servers and browsers, thereby obtaining plaintext content from encrypted communications. SSL proxy certificates are certificates re-signed using the device's own certificate for the Web server certificate. The entire process is shown in Figure 2 [Figure 2: see original paper].

The second method deploys MITM-supporting proxy programs such as Fiddler, Charles, and whistle on data analysis servers to replace original packet capture programs for grabbing and analyzing data packets. Using Charles proxy program as an example, its HTTPS parsing process is shown in Figure 3 [Figure 3: see original paper]. This method is essentially still based on gateway log collection and analysis mode, only replacing the original non-HTTPS-parsing gateway

device with an HTTPS-parsing gateway device, still requiring log harvesting programs to generate usage statistics reports. Due to auditing requirements, more gateway devices are beginning to support SSL proxy functions.

As shown in Figure 3, the Charles proxy program masters the server certificate public key and HTTPS connection symmetric key throughout the communication process, enabling decryption of all ciphertext using corresponding keys. The entire communication process is transparent to Charles. By adding monitoring code to MITM-supporting proxy programs, collection of HTTPS-based electronic resource access data can be achieved.

3.3 Data Diversion Methods

Both SSL proxy-supported gateway devices and proxy programs use MITM technology to parse HTTPS. MITM technology requires direct communication with both client and server, necessitating that devices or proxy programs using MITM technology be integrated into the existing network topology. For data analysis servers deploying proxy programs, if directly connected to the network topology, they must handle not only electronic resource access data but also non-electronic resource traffic. Since data analysis servers are not dedicated gateway devices, excessive load may cause server malfunction. To solve this problem, data diversion methods can be used to forward only electronic resource access data to the data analysis server while sending non-electronic resource access data directly to the exit gateway.

The most common packet forwarding technology is Policy-Based Routing (PRB), which is a routing selection mechanism based on user-defined strategies. By enabling policy routing on core switches, creating Access Control Lists (ACLs) storing electronic resource server IP addresses, and configuring forwarding policies, electronic resource access data diversion can be achieved. When network access data reaches the core switch, the destination IP address is matched against the ACL list. Packets with electronic resource server IP addresses as destinations are forwarded to the data analysis server, while other access data is sent directly to the exit gateway. Policy routing's advantage is that packet forwarding occurs on core switches without user involvement, making the change imperceptible to users.

Policy routing has certain limitations: its forwarding decisions are based on destination server IP addresses. Some electronic resources use CDN (Content Delivery Network) technology to accelerate access by deploying multiple cache servers distributed to regions or networks with concentrated user access. When users access the website, global load balancing technology directs users to the nearest functioning cache server, which directly responds to user requests. Electronic resources using CDN technology have multiple cache servers and corresponding multiple IP addresses. When the number of IP addresses for all electronic resources exceeds the maximum supported by ACL lists, some electronic resource access data cannot be forwarded to the data analysis server, resulting

in inaccurate usage statistics.

Another data diversion method is PAC (proxy auto-config), an automatic proxy configuration script file that determines whether browsers access network resources through the default channel or proxy server channel. Browsers achieve automatic proxy functionality by configuring PAC files. PAC files contain a JavaScript function FindProxyForURL that returns a string containing one or more access rules determining whether browsers access network resources through the proxy program. In the FindProxyForURL function, the browser's accessed URL is evaluated. When the URL contains electronic resource domain information, it is specified to access through the proxy program, thereby diverting electronic resource access traffic. Compared to policy routing, PAC uses domain information for data diversion, requiring significantly fewer matching rules and no configuration on gateway devices. PAC is more suitable as a data diversion method, but requires users to configure PAC files in their browsers, which may cause user resistance.

4 Comparative Analysis of Three HTTPS Parsing Methods

The three methods (hereinafter referred to as browser extension, SSL gateway, and proxy program) can all parse HTTPS-based access data, but each method applies to different environments. This section compares these methods from four aspects: hardware/software requirements, network conditions, system functionality, and user cooperation needs.

4.1 Hardware and Software Requirements

Hardware and software requirements refer to hardware devices, installed software, and self-developed programs and files needed for each method. In terms of hardware, all three methods require a dedicated server to store electronic resource access records and generate usage statistics reports. SSL gateways additionally require purchasing dedicated gateway devices, representing the highest cost. Proxy programs need to analyze and forward data packets, requiring higher server performance. Hardware costs depend on traffic volume but generally do not exceed SSL gateways. Browser extensions only require a Web server on the server side to receive information collected by browsers, with minimal server performance requirements and the lowest hardware cost.

In terms of software, all three methods require database programs to store user request information. SSL gateways need log harvesting programs; browser extensions require extension programs and Web pages to receive access request information; proxy programs need corresponding proxy software with code to record electronic resource access requests; proxy programs using data diversion also require configuring core switch policy routing functions or creating PAC files. In terms of technical difficulty, proxy programs require the highest technical skill, followed by browser extensions, with SSL gateways being the lowest.

4.2 Network Conditions

Network conditions refer to whether methods require changes to existing network topology and where servers or gateway devices must be installed in the network. SSL gateways have the highest network environment requirements, needing to be directly integrated into the existing network topology as a core node. Proxy programs not using data diversion, like SSL gateways, must be integrated into the network topology as core nodes. Proxy programs using policy routing for data diversion need data analysis servers connected to core switches with direct communication capability, without any additional network nodes in between. Proxy programs using PAC diversion can theoretically place servers anywhere within the organization's internal network as long as user access data can be forwarded to the server. However, to ensure access speed and reduce intermediate nodes, servers should still be directly connected to core switches. Browser extensions have the lowest network environment requirements—the server only needs to receive information sent by users and can even be deployed on external networks.

4.3 System Functionality

System functionality can be evaluated from three aspects: data collection completeness, usage statistics report generation timeliness, and system control capability. Regarding data collection completeness, SSL gateway data comes from gateway logs, and completeness depends on gateway log information volume. Typically, gateway logs only record URL, source IP, and access time—limited information resulting in poor data collection completeness. Both proxy programs and browser extensions can directly obtain complete access behavior data, including browser type, user IP address, destination IP address, access time, accessed page URL, accessed page HTML content, downloaded resource type, and even user name and department information through user IP addresses, resulting in highly complete data collection.

Regarding usage statistics report generation timeliness, SSL gateway report generation frequency depends on log harvesting program collection frequency, typically harvesting logs daily or hourly—timeliness at the hour level. Proxy programs and browser extensions record access data in real-time as users access information, enabling real-time monitoring of user access behavior and real-time generation of usage statistics reports—timeliness at the real-time level.

Regarding system control capability, SSL gateways generally only record access behavior without data content modification capability. Some SSL gateways provide blacklist functions to block access to specified domains, representing average system control capability. Proxy programs and browser extensions have complete control over user access behavior, capable of intercepting or modifying user requests and server response content, representing very strong system control capability. Browser extensions must be installed on user browsers, and their functions must consider universality issues, making it difficult to implement spe-

cific function control for individual users. When functions change, they must wait for users to update browser extensions before new functions take effect. Proxy program control over user requests occurs on data analysis servers, allowing more functions to be set for different situations and different response strategies for specific user requests, with system function updates taking immediate effect. Therefore, proxy programs have higher system control capability than browser extensions.

4.4 User Cooperation Requirements

User cooperation requirements refer to client-side actions needed from users, such as installing certificates, installing browser extension programs, modifying system settings, etc. Both MITM-based SSL gateways and proxy programs require sending SSL proxy certificates to user browsers for data encryption. Typically, these certificates are self-signed by SSL gateways or proxy programs. After receiving certificates, browsers validate them. When browsers discover certificates are not issued by trusted root certificate authorities, they display security certificate trust warnings when accessing relevant domains, potentially preventing normal access to corresponding electronic resources. To solve this problem, users must import the SSL proxy certificate's root certificate into their browser and add it to trusted root certificate authorities. Since users manually trust the SSL proxy certificate's root certificate, the corresponding SSL proxy certificate will be trusted by the browser, allowing normal electronic resource access.

Proxy programs using PAC diversion also require users to configure PAC-related settings in their local browsers to enable automatic proxy functionality. Browser extensions require users to install extension programs in their browsers and enable extension functionality for proper operation. All three methods require user cooperation to achieve collection of HTTPS-based electronic resource data.

The comparative analysis of the three methods is summarized in Table 2 .

Table 2 Comparative Analysis of Three Methods

Comparison Item	SSL Gateway	Browser Extension	Proxy Program
Software Technical Difficulty	Low	Medium	High
Hardware Requirement Cost	High	Low	Medium
Network Environment Requirements	High	Low	Medium
User Cooperation Requirements	Certificate Installation	Extension Installation	Certificate Installation

5 Implementation and Application

5.1 Selection Strategy for Three Methods

SSL gateways require adding or replacing gateway devices, changing the original network topology, and have the highest hardware cost but lowest system functionality. If libraries have low requirements for electronic resource usage statistics reports and their institution happens to need gateway device upgrades, they can recommend that network departments select SSL proxy-supported gateway devices.

Browser extensions have the lowest cost and provide system functionality that meets most library needs. However, they require installing extension programs in client browsers, and some users may refuse installation due to security concerns. When libraries have strong control over users and can mandate browser extension installation, and when budgets are limited, the browser extension method can be adopted.

Proxy programs provide the strongest system functionality but have the highest technical requirements and also require cooperation from network departments and users. If libraries require very strong system functionality, need systems that can adopt different collection and control strategies for different users, require detailed and accurate usage statistics reports, have certain technical development capabilities, and can obtain user cooperation, they can adopt the proxy program method.

Taking my institution, the Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences (hereinafter referred to as "our institute"), as an example, the following factors were considered when building the electronic resource usage statistics system: (1) Our institute requires research groups to share part of electronic resource costs based on usage volume, demanding that the system provide accurate statistics for individuals and research groups; (2) The system must monitor, warn, and handle abnormal literature downloading behavior; (3) Users can accept installing SSL proxy certificates but cannot accept browser extension programs and PAC files; (4) Our institute's core switches and firewall devices support policy routing functions, and the network department is willing to cooperate in configuring policy routing. Based on these factors, our institute selected the Fiddler proxy program to parse and record electronic resource access data and enabled policy routing on firewall devices to divert electronic resource access data.

5.2 Implementation Effects of Our Institute's Electronic Resource Usage Statistics System

Based on the above solution, our institute completed deployment of the localized electronic resource usage statistics system (hereinafter referred to as "the system") in July 2017. After multiple modifications and adjustments, the system now operates stably and provides good service.

Regarding data collection, the system has achieved full-text access data recording for nine databases: ACS, Wiley, ScienceDirect, RSC, Nature, Science, AIP, OSA, and Springer, storing 431,342 full-text access records in 2018. This data can generate full-text download volumes, download proportions, and cost-per-article metrics for each database, providing strong support for libraries to understand electronic resource usage and adjust resource guarantee strategies. Additionally, it helps understand research groups' disciplinary directions and literature needs, assisting libraries in developing personalized information services.

Regarding data application, based on full-text access data from December 1, 2017 to November 30, 2018, our institute completed research group cost-sharing for electronic resources in December 2018. A total of 92 research groups shared electronic resource costs of 1,002,090.27 yuan. Since the system's full-text access data includes user network accounts, IP addresses, full-text URLs, and access times, the detailed and accurate data resulted in no objections from research groups regarding data and costs, enabling smooth implementation of cost-sharing.

Regarding system control, when users click article titles in the ACS database, they automatically jump to full-text pages generating one full-text access record. When users then download PDF files on full-text pages, another full-text access record is generated, causing duplicate full-text access records. To address this, the system implemented URL replacement measures in the Fiddler program. When the system detects users clicking ACS titles, it replaces the full-text page URL with the corresponding abstract page URL. When users click ACS database article titles, they no longer access full-text pages but abstract pages, avoiding unnecessary duplicate full-text data access.

5.3 Existing Problems and Future Development Directions

Both SSL gateways and proxy programs require user certificate installation, browser extensions require user extension installation, and PAC diversion methods require user browser configuration. Regardless of the method, all require users to add extra files on the client side, inevitably infringing on user privacy and posing data security risks. Users generally find it difficult to accept installing additional programs on the client side due to data security and personal privacy concerns. In applying localized electronic resource usage statistics systems, libraries should consider not only how to protect user privacy from infringement but also ensure that installing certificates or extensions does not create risks of user data leakage.

Localized electronic resource usage statistics systems should consider introducing new technical methods in the future to achieve HTTPS-based electronic resource access data collection without requiring users to install any programs. Collected data should not only be used to generate usage statistics reports or visualize data display but should employ data analysis, machine learning, and deep learning methods for in-depth mining, analyzing user needs, building user

profiles, and providing foundations for personalized knowledge services.

References

[1] Chen Daqing, Ye Lan, Yang Wei, et al. Design and implementation of electronic resource usage statistics platform USSER[J]. Library and Information Service, 2015, 59(1): 106-112.
[2] Anderson E K. Electronic resource management systems and related products[J]. Library Technology Reports, 2014, 50(3): 30-41.
[3] Wang Dandan. Research on the current situation and trends of digital library user usage data statistics[J]. Library Development, 2012(11): 66-69.
[4] Tripathi M, Jeevan V. A selective review of research on e-resource usage in academic libraries[J]. Library Review, 2013, 62(3): 134-156.
[5] Zhu Ling, Cui Haiyuan. Discussion on evaluation methods for data acquisition quality of electronic resource usage monitoring and statistics systems in university libraries[J]. Library and Information Service, 2016, 60(5): 51-56.
[6] Yan Xiaodi, Shao Jing, Zhou Qi, et al. Design and implementation of electronic resource usage statistics gateway system[J]. New Technology of Library and Information Service, 2008, 24(8): 97-100.
[7] Wang Xiaoliang, Wang Wei. Constructing electronic journal database statistical analysis system through firewall log mining[J]. New Technology of Library and Information Service, 2013, 29(S1): 122-126.
[8] Guo Zhenying, Zhao Wenbing, Wei Yuhui. Analysis and design of electronic resource log statistics system[J]. New Technology of Library and Information Service, 2008, 24(9): 102-106.
[9] Zhou Xin, Lu Kang. Research on reader behavior data mining based on library digital resource access system[J]. Journal of Modern Information, 2016, 36(1): 51-56.
[10] Zhang Jilong, Yin Shenqin, Chen Tie. Research on user information behavior data collection methods based on ERU: A case study of Fudan University Library[J]. Library Journal, 2014, 33(12): 10-16.
[11] Zou Rong, Zhang Chengyu, Jiang Airong, et al. Design and application of electronic resource access management and control system[J]. Library and Information Service, 2010, 54(1): 121-124.
[12] Shi Xiaohua, Qian Yin, Xie Rui. Design and application of university electronic resource access management and control system[J]. Application Research of Computers, 2011, 28(3): 1042-1045.
[13] Wang Zhengjun, Dong Xiaomei, Yu Xiaoyi. Design and implementation of digital resource evaluation system based on bypass monitoring[J]. Library and Information Service, 2015, 59(9): 52-57.
[14] Gourley D, Totty B. HTTP: The definitive guide[M]. Translated by Chen Juan, Zhao Zhenping. Beijing: Posts & Telecom Press, 2012.
[15] Yang A. Detailed explanation of SSL[EB/OL]. [2018-11-07]. https://www.cnblogs.com/NathanYang/p/918
[16] Qu J. Introduction to three methods for decrypting HTTPS traffic[EB/OL]. [2018-11-07]. https://imququ.com/post/how-to-decrypt-https.html.
[17] fkyq01. webRequest[EB/OL]. [2018-11-07]. https://developer.mozilla.org/zh-

CN/docs/Mozilla/Add-ons/WebExtensions/API/webRequest.
[18] SSL Proxy[EB/OL]. [2018-11-07]. http://docs.hillstonenet.com/cn/Content/9_{Security}/SSLProxy.htm.
[19] 123frea321. CDN high-speed cache server setup and configuration[EB/OL]. [2018-11-07]. https://blog.csdn.net/zxy15771771622/article/details/79310601.

A Comparative Study of Localized Electronic Resource Usage Statistics System for Resolving HTTPS Access Data

Chen Guang
Fujian Institute of Research on the Structure of Matter, Chinese Academy of Sciences, Fuzhou 350002

Abstract: [Purpose/significance] Aiming at the new problems of localized electronic resource usage statistics system, this paper proposes methods for resolving HTTPS access data, analyzes and evaluates these methods, and provides reference for libraries to solve the collection problem of electronic resource access data based on HTTPS protocol in localized usage statistics systems. [Method/process] From four aspects of hardware and software requirements, network conditions, system functions, and user cooperation requirements, this paper compares and evaluates three methods including browser extensions, proxy programs supporting MITM technology, and devices supporting SSL proxy gateway. [Result/conclusion] The study shows that proxy programs supporting MITM technology are moderately costly and have the strongest system functions, making them most suitable for localized electronic resource usage statistics systems. On the basis of solving HTTPS access data collection, how to ensure user privacy and data security while obtaining user cooperation and coordination will be the biggest difficulty in the application of localized electronic resource usage statistics systems.

Keywords: electronic resource; usage statistics; browser extension; proxy; SSL proxy gateway

---

Contents for Next Issue

Review of Personal Information Protection Research in the Big Data Era (Jiang Panpan)

Note: Figure translations are in progress. See original paper for figures.